CS-4341/6341: Project Review Form

***Instructions**:Fill out this form before class and then one person in your team emails its reviews to the instructor (as PDF attachments called review1.pdf and review2.pdf) . You will fill out one form per review. Do not share this review with any other team. Your review is confidential – you will not be identifed as reviewers to the team whose project you are reviewing.*

Your (review) team letter: P

Name of zip file (of the project you are reviewing):

Aditya-Huzefa-Jerome.zip

1. Describe in a few sentences (your own words) the main idea in this project.

   This is a project that imitate the process of thermal conduction. There are some equations about the thermal conduction and eventually derive an equation to update the current state from previous state. Also, there is a graph help us to understand what is 'Steady State, Transient, Relativistic and Quantum'.

2. Were you able to get the demo working? If not, describe the issues faced.

   Yes, it works.

3. Rate from 1 to 5 (1=weak, 5=strong) the following:

   a. The quality of description of the main idea: 5
   b. The ease with which the demo compiled and ran: 5
   c. The extent to which the demo aligned with the main idea: 3
   d. The quality of description of the exercise: 5
   e. The appropriateness of the exercise to the main idea: 5
   f. The extent to which you learned by doing the exercise: 3
   g. Average rating (average of ratings in a through f): 4

4. In the space below, write your solution to the exercise, and bring along code (to class) to demonstrate completion of the exercise. If you could not complete the exercise, explain why. Use additional pages if needed.

```
boolean updateMatrix() {
    // Create a clone of matrix
    double[][] clone = getClone(matrix);

    // Update using the implicit method
    for (int j = 1; j < H - 1; j++) {
      for (int i = 1; i < L - 1; i++) {

                                  // TODO:
```

```
                                        // Implement the equation and update the matrix accordingly


                                        matrix[j][i]=clone[j][i]+rx*(clone[j][i+1]-
2*clone[j][i]+clone[j][i-1])+ry*(clone[j+1][i]-2*clone[j][i]+clone[j-1][i]);


          }
      }

      // Check for maximum corresponding difference, return false if it is less than epsilon
      double e = getMaxCorrespondingDifference(matrix, clone);
      if (e <= epsilon && e != -Double.MAX_VALUE)
         return false;
                return true;

   }

   double getMaxCorrespondingDifference(double[][] matrix, double[][] clone) {
      double maxError = -Double.MAX_VALUE;

                // TODO:
      // calculate and return the maximum difference between the corresponding values of
matrix and clone
                // refer to the pseudocode for help
                for (int i=0;i<L;i++)
                {
                        for (int j=0; j<H; j++)
                        {
                                double e=Math.abs(clone[i][j] - matrix[i][j]);
                                if (e > maxError && e != 0.0) maxError = e;

                        }
                }




      return maxError;
   }

}
```

```
0.0 100.0 100.0 100.0 100.0
0.0 49.99998094683869 62.49997305476086 49.99998094683869 0.0
0.0 37.49997305476087 49.99996189367738 37.49997305476087 0.0
0.0 49.99998094683869 62.49997305476086 49.99998094683869 0.0
0.0 100.0 100.0 100.0 100.0
```

2.

```
0.0 100.0 100.0 100.0 100.0
0.0 49.999984911255545 62.49997866129294 49.999984911255545 0.0
0.0 37.49997866129296 49.99996982251109 37.49997866129296 0.0
0.0 49.999984911255545 62.49997866129294 49.999984911255545 0.0
0.0 100.0 100.0 100.0 100.0
```